ISSN 2277-2685
IJESR/April-June. 2026/ Vol-16/Issue-2/1104-1109
K Akhila *et. al.*, / International Journal of Engineering & Science Research
Full Length Article

Web-Based Food Ordering System

Ms K Shireesha[1], K Akhila[2], M Laxmi Prasanna[3], B Manasa[4]
[1]Associate Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India
[2,3,4]B.Tech Students; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India
Mail Id: kethavathakhila6768@gmail.com[2]
Accepted 13-06-2026

Abstract
The Web-Based Homemade Food Ordering System is a full-stack web application designed to bridge the gap between homemaker food providers and local consumers seeking fresh, hygienic, and affordable homemade meals. The platform enables homemakers to register, manage profiles, and list food items, while users can conveniently browse, search, and filter available food based on categories such as vegetarian, non-vegetarian, tiffins, and snacks.
The system integrates location-based services (LBS) to display nearby food providers, improving accessibility and enhancing user experience. It also supports pre-ordering and subscription-based meal planning, allowing users to place advance orders for daily, weekly, or monthly food requirements.
The application is developed using React.js for the frontend and Node.js with Express.js for backend services, along with a MySQL database for efficient data storage and management. The system follows a scalable client-server architecture to ensure performance, reliability, and maintainability.
The primary objective of this project is to empower home-based food entrepreneurs, especially homemakers, by providing them with a digital platform to sell homemade food, while simultaneously offering users a convenient, cost-effective, and healthy alternative to restaurant-based fast food services.

Keywords: *Homemade Food System, Web Application, React.js, Node.js, Express.js, MySQL, Location-Based Services, Food Ordering Platform, Subscription System, Pre-order System, Homemaker Empowerment, Online Food Delivery.*

Introduction

The increasing demand for convenient and healthy food options has led to the rapid growth of online food delivery systems. However, most existing platforms primarily focus on restaurant-based services and instant delivery models, leaving limited opportunities for homemaker women who possess strong culinary skills but lack access to structured employment channels. At the same time, students and working professionals often struggle to obtain affordable, hygienic, and home-cooked meals that match their dietary preferences and budget requirements. This gap highlights the need for a dedicated system that connects homemakers directly with consumers while ensuring quality, affordability, and accessibility.

Existing food delivery applications such as Zomato and Swiggy are designed mainly for restaurant-based food services and do not adequately support homemade food providers. As a result, homemakers are unable to utilize these platforms to sell their food products, and users find it difficult to access authentic homemade meals. Additionally, these platforms lack features such as pre-ordering, subscription-based meal plans, and transparent pricing models for home-cooked food services. Informal methods of food selling, such as phone calls or messaging applications, are also widely used but lack proper organization, reliability, and security.

The absence of a structured system creates several challenges for both users and homemakers. Instant food ordering models often lead to increased costs and reduced nutritional value due to reliance on processed or restaurant-prepared meals. Moreover, the lack of a pre-order mechanism contributes to food wastage and reduces efficiency in meal preparation. Homemakers are also restricted by the absence of digital platforms that can help them expand their services and generate stable income. Pricing inconsistency and the absence of subscription-based models further reduce transparency and user convenience.

To address these limitations, the proposed system introduces a web-based Homemade Food Ordering Platform that directly connects homemaker women with nearby users. The system is designed around a pre-order model, allowing users to select meals in advance based on location and availability. Once an order is confirmed, homemakers prepare food accordingly, ensuring freshness and minimizing wastage. The platform also supports flexible weekly and monthly subscription plans, enabling users to access consistent meal services at affordable prices.

Additionally, the system integrates a chatbot-based assistance feature to simplify ordering, subscriptions, and user support, while also offering flexible delivery and pickup options.

The proposed system provides multiple advantages over existing solutions. It creates income-generating opportunities for homemaker women by enabling them to operate as independent food providers. Users benefit from access to healthy, hygienic, and cost-effective homemade meals. The pre-order-based structure ensures food freshness and significantly reduces wastage. Subscription models offer transparent and predictable pricing, improving user trust and affordability. The inclusion of a chatbot enhances usability by guiding users through ordering and support processes. Furthermore, the system supports flexible delivery options, strengthens local community engagement, promotes sustainable food consumption practices, and offers a user-friendly interface with potential multilingual support.

Requirements Analysis

Computational Resources

Hardware Requirements

The system requires a minimum hardware configuration consisting of an Intel i5 processor, 4 GB RAM, and 200 GB of hard disk storage. This configuration is sufficient to support development, testing, and basic deployment of the application.

Software Requirements

The proposed system operates on Windows 10 or higher operating systems. Development is carried out using Visual Studio as the primary tool. The front-end is built using HTML, CSS, JavaScript, and Bootstrap for responsive design. The back-end is developed using Node.js, while MySQL 8.4 is used for database management and data storage.

Life Cycle Model

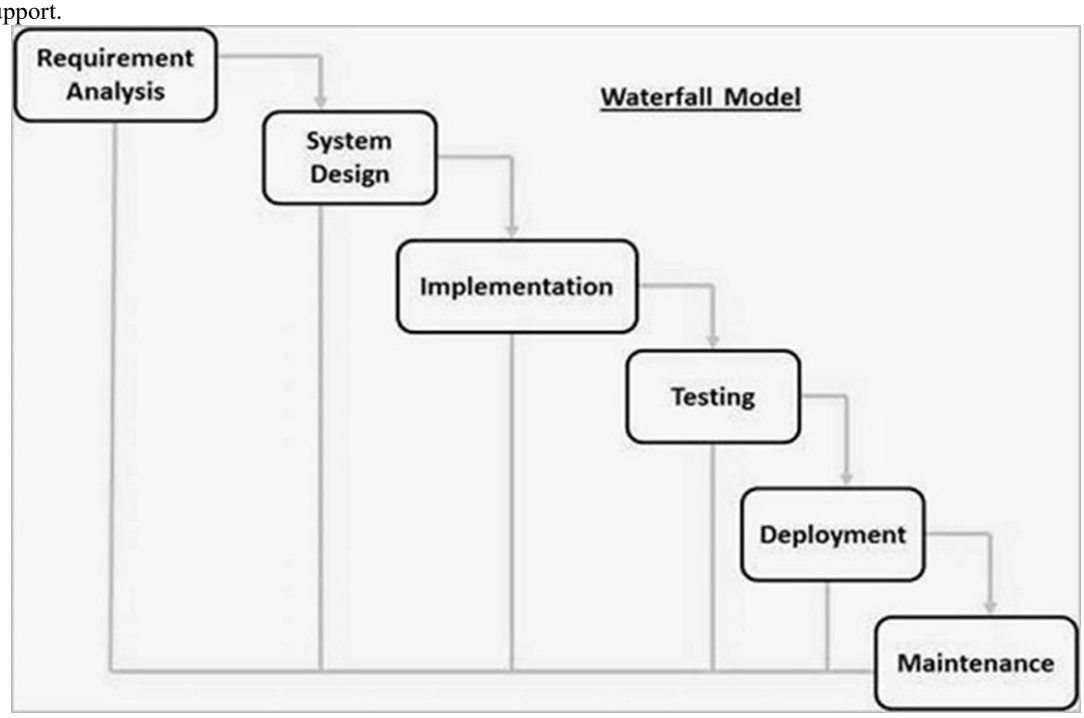


Fig 1: Waterfall Model

The development of the proposed system follows the Waterfall Model, which is a linear and sequential software development life cycle approach. This model is suitable for projects where requirements are well-defined and stable throughout the development process.

In the Requirement Analysis phase, system requirements were gathered and analyzed to define the core functionalities of the platform, including user registration, login, location-based search, food ordering, and subscription management.

During the System Design phase, the overall architecture of the system was planned. UML diagrams such as Use Case, Class, Sequence, and Activity diagrams were developed to represent system behavior and structure. In addition, the user interface layout and database schema were designed to ensure smooth system interaction.

In the Implementation phase, the system was developed using HTML, CSS, and JavaScript for the front-end, along with Node.js for backend functionality. Multiple pages such as Home, Login, Registration, Pre-order, Subscription, and Payment were implemented to support user operations.

The Testing phase involved validating all system functionalities, including authentication, food filtering, ordering, subscription services, and payment processing. Errors identified during testing

were corrected to ensure system stability and correctness.

In the Deployment phase, the system was prepared for hosting on a web server, enabling users to access the platform remotely and utilize its services.

Finally, in the Maintenance phase, future enhancements are considered, such as integrating real-time payment gateways, adding user reviews and ratings, implementing notification systems, and extending support to mobile applications to improve overall usability and functionality.

Design

Architectures

Project architecture represents number of components we are using as a part of our project and the flow of request processing i.e. what components in processing the request and in which order. An architecture description is a formal description and representation of a system organized in a way that supports reasoning about the structure of the system. Architecture is of two types. They are

(1) Software Architecture

(2) Technical Architecture

Software Architecture

Fig 2: Software Architecture

Technical Architecture

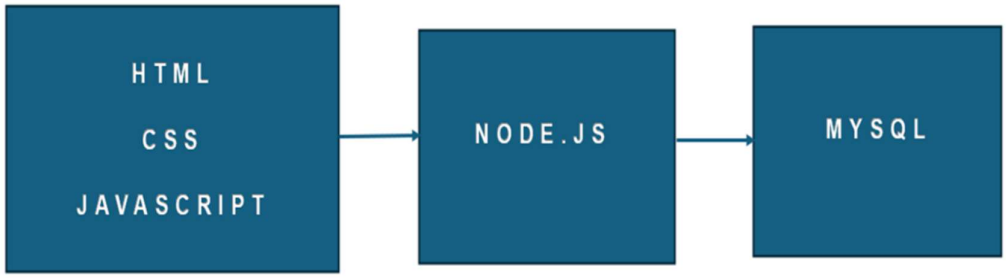


Fig 3: Technical Architecture

Implementation

The proposed awareness platform is developed using a combination of modern web technologies to ensure a responsive, interactive, and user-friendly experience. The system primarily utilizes HTML, CSS, and JavaScript for front-end development, while MySQL is used as the backend database for efficient data storage and retrieval. These technologies collectively support a scalable and maintainable web application architecture.

HTML (HyperText Markup Language) is used to define the structural layout of the web pages. It organizes the content of the application into meaningful sections such as headers, navigation

menus, forms, buttons, images, and hyperlinks. In this project, HTML forms the foundation for multiple user interfaces including the Home page, Login page, Registration page, About page, Pre-Order page, Subscription page, and Payment page.

CSS (Cascading Style Sheets) is applied to enhance the visual presentation and user experience of the platform. It is responsible for controlling layout design, color schemes, typography, spacing, and responsiveness. In this system, CSS is extensively used to style navigation bars, food item cards, buttons, input forms, and background elements, ensuring an attractive and consistent interface across all pages.

JavaScript is incorporated to introduce dynamic behavior and interactivity within the application. It enables real-time user interaction such as filtering food items based on categories like vegetarian, non-vegetarian, breakfast, snacks, and handling user actions like selections and confirmations. JavaScript also improves responsiveness by updating content without requiring full page reloads.

HTML provides a well-structured framework for organizing web content using semantic elements and tags. It ensures cross-platform compatibility, allowing web pages to render consistently across different browsers and devices. Its simple syntax makes it easy to learn and implement, even for beginners. Additionally, HTML supports accessibility features through semantic tags such as header, nav, and footer, as well as attributes like alt for images, improving usability for differently-abled users. Furthermore, HTML is highly versatile and supports multiple content types including text, images, videos, tables, and forms, enabling the development of rich and interactive web applications.

System Logic (Pseudocode Description)

The system logic of the application is designed to manage user authentication, location-based food recommendations, filtering mechanisms, and dynamic content rendering. The application begins by initializing the user interface and checking the authentication status of the user. Based on this status, the system either allows access to food-related services or prompts the user to log in.

Once the user is authenticated, the system captures the user's geographic location using browser-based geolocation services. This location data is then utilized to fetch nearby food items from the backend server through API calls. Each food item retrieved from the database is processed to calculate the distance between the user and the vendor location using a distance computation function. In addition, food attributes such as category (e.g., snacks, tiffins, meals) and type (vegetarian or non-vegetarian) are normalized for consistency.

After processing, the system applies multiple filtering operations. Users can filter food items based on search keywords, category selection, or dietary preferences. The filtered results are then sorted based on proximity, ensuring that the nearest available food options are displayed first. This enhances user convenience and improves delivery efficiency.

Furthermore, the system estimates delivery time dynamically by analyzing the calculated distance between the user and the vendor. Shorter distances correspond to reduced delivery time estimates, while longer distances increase the estimated time accordingly.

Finally, the user interface renders the complete layout, which includes the navigation bar, hero section with search functionality, category filters, and dynamically generated food cards. Depending on the system state, different views are displayed: a loading skeleton during data retrieval, an error message in case of failure, a list of available food items when data is successfully fetched, or an empty state message when no results are found. If the user is not logged in, the system restricts access and displays a login prompt. Additional components such as a chatbot assistant and footer are consistently rendered across the application to enhance usability and support.

Testing

Software testing is a critical phase in the software development lifecycle that evaluates an application to determine whether it satisfies the specified functional and non-functional requirements. The primary objective of testing is to identify defects and ensure that the system operates reliably, efficiently, and as expected under different conditions. In the context of modern digital systems, where users increasingly depend on online services such as banking, e-commerce, and food ordering platforms, the reliability of software becomes highly important. Even minor defects in such systems can lead to significant financial losses, security vulnerabilities, and reduced user trust.

Therefore, software testing plays a vital role in delivering high-quality applications by ensuring correctness, stability, and security. It also contributes to customer satisfaction by verifying that the system meets user expectations. Additionally, testing helps in identifying potential security risks and improving the overall product quality before deployment.

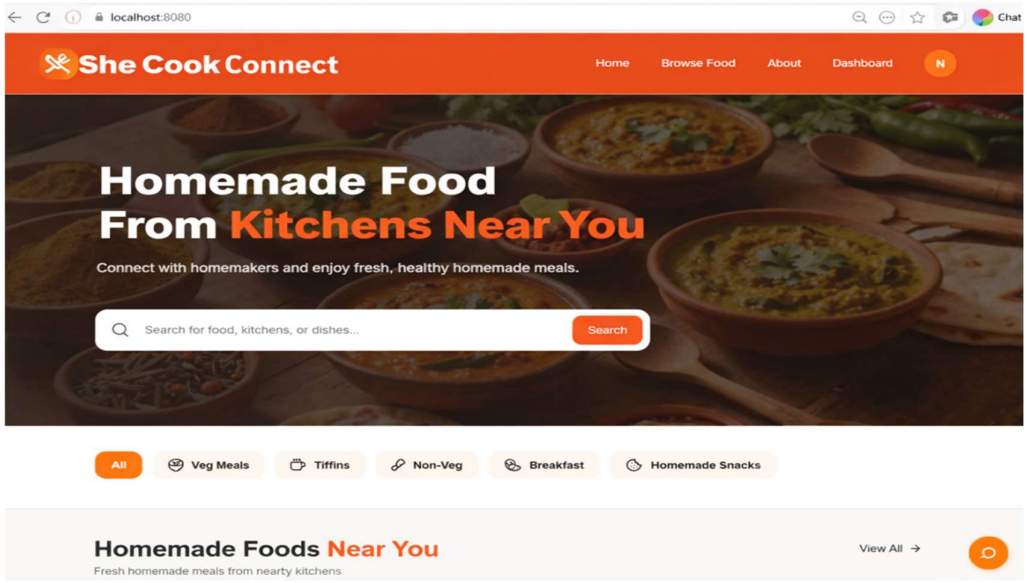
Dimensions of Testing

Software testing can be analyzed through multiple dimensions to ensure comprehensive validation of the system. These dimensions include different layers of the application such as the database layer, application programming interfaces (APIs), and user interface (UI). Each layer is tested to ensure proper communication and functionality within the system architecture.

Testing also varies based on scale, including unit-level testing, module-level testing, integration

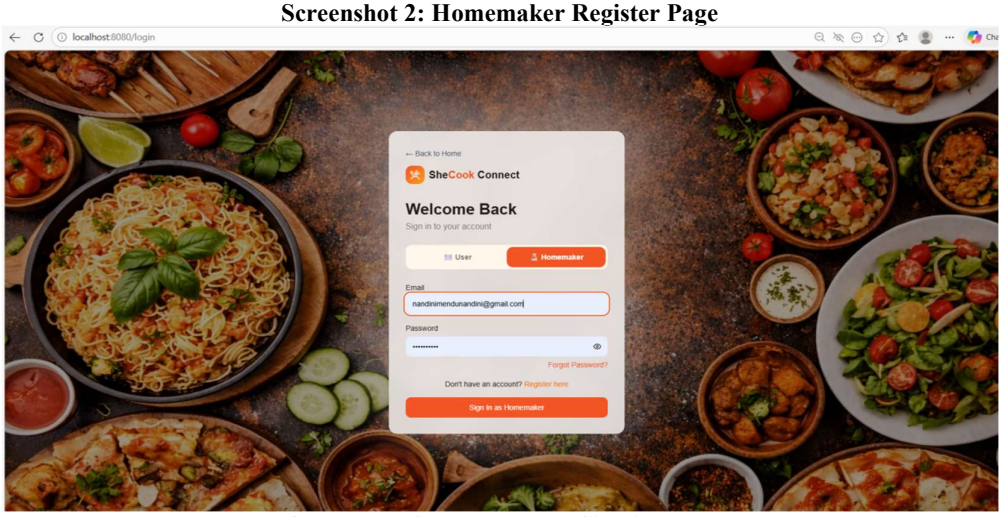
testing, and end-to-end scenario testing. Furthermore, testing can be categorized based on its type, such as functional testing, performance testing, and security testing. Another important dimension is the testing methodology, which may include manual testing, exploratory testing, or automated testing depending on project requirements and complexity.

Screenshots

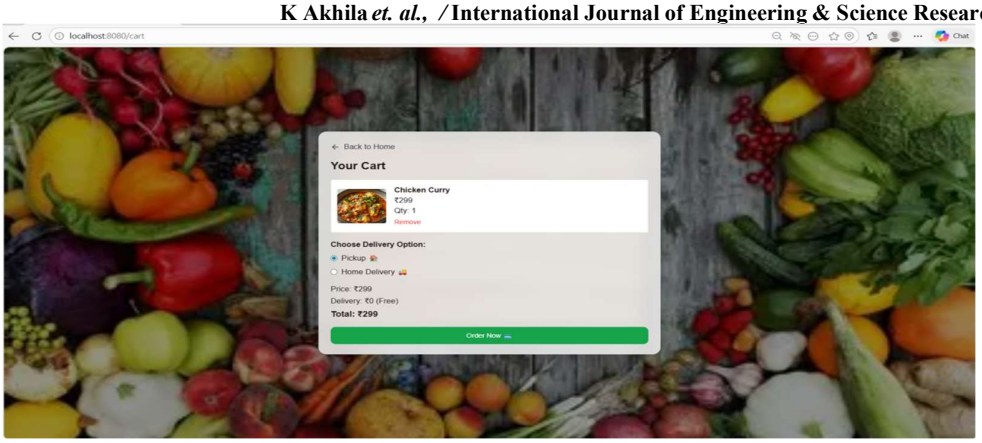


Screenshot 1: Home Page

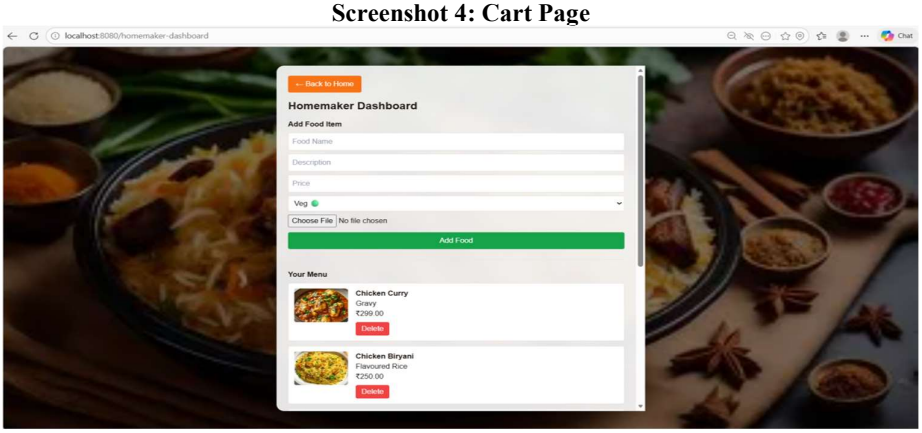
Screenshot 2: Homemaker Register Page



Screenshot 3: Homemaker Sign in Page



Screenshot 4: Cart Page



Screenshot 5: Dashboard Page

Conclusion

This platform introduces an innovative, user-friendly solution for connecting homemakers with food seekers. By providing a pre-order system, women empowerment, and customizable meal plans, it goes beyond what current food delivery apps offer, creating a supportive, local community that brings homemade meals to people's doorsteps. Through features like chatbot assistance, item-based subscriptions, and safety-focused design, it creates a new kind of food-sharing experience that is both sustainable and trustworthy.

Future Scope

In the future, the system can be developed as a mobile application to make homemade food ordering more convenient and accessible for users. Additional features like online payment integration, GPS-based delivery tracking, and AI-based food recommendations can also be added. The platform can further expand to multiple cities, connecting more homemakers with customers and increasing employment opportunities.

References

[1] Gupta, R., & Kumar, S. (2024). *Addressing food insecurity through local solutions*. International Journal of Sustainable Development, 15(3), 112–125.

[2] Statista. (2024). *Food delivery industry report*. Available at: https://www.statista.com

[3] Forbes. (2023). *The rise of homemade food delivery platforms*. Available at: https://www.forbes.com

[4] United Nations. (2024). *Sustainable Development Goals*. Available at: https://www.un.org

[5] Patel, P. (2025). *From home to market: Building a sustainable food business*. EcoPress, San Francisco